CLAIMS

Sub A1

We claim:

1. ~~A method of maintaining application program components on a network comprising:~~

~~maintaining on a server the application program, the program including components, each having a version identification, and maintaining a catalog of components with the version identifications;~~

~~maintaining the application program on a client;~~

~~in response to a call to the server from the client, causing the server to download the catalog to the client and, in the client, comparing the version identification between the components maintained on the server, indicated in the catalog, and the components maintained on the client;~~

~~updating the application program components on the client by downloading from the server to the client the selected components for which the version identifications do not match and replacing the selected components on the client; and~~

~~executing the updated application program on the client.~~

2. A method as claimed in Claim 1 further comprising storing in a persistent cache on the client a portion of the catalog which includes the components of the updated application program on the client.

3. A method as claimed in Claim 1 further comprising updating the application program on the client by downloading from the server to the client the selected components not present on the client.

18

Sub C7

~~4. A method as claimed in Claim 1 wherein the server download the catalog and components to the client for updating the application program on the client by way of hypertext transfer protocol.~~

~~5. A method as claimed in Claim 1 wherein the server download the catalog and components to the client for updating the application program on the client by way of file transfer protocol.~~

6. A method as claimed in Claim 1 wherein the catalog maintained on the server includes network addresses of further servers from which the components can be retrieved.

7. A method as claimed in Claim 1 wherein the catalog maintained on the server includes directory locations on the client in which the downloaded components are stored for proper execution of the application program.

8. A method as claimed in Claim 1 wherein the call to the server is transmitted to the server by a launcher on the client which operates as a proxy to the application program such that selecting the application from the client to execute the program engages the launcher to communicate with the server to cause the server to download the catalog, update the application program on the client, and execute the updated application program on the client.

9. A method as claimed in Claim 8 wherein the launcher is a functional component of an operating system running on the client.

10. A method as claimed in Claim 1 wherein the catalog includes procedures for executing supplemental programs on the client prior to executing the updated application program.

11. A method as claimed in Claim 10 wherein the supplemental programs are executed following the execution of the application program.

12. A method as claimed in Claim 10 wherein the supplemental programs include a virus scanning program.

13. A method as claimed in Claim 1 further comprising specifying a maximum wait-time interval to limit any delay associated with updating the application program, and in a catalog maintained on the client specifying a list of further servers on the network, each including a copy of the catalog, such that when the server fails to download the catalog within the maximum wait-time interval, the client cancels the download and routes the call to one of the further servers to engage a new download and so on until the catalog has been downloaded within the specified maximum wait-time interval.

14. A method as claimed in Claim 1 further comprising:

specifying in the catalog a cryptographic digest for each component to ensure authenticity and integrity of the component;

updating the application program on the client by downloading from the server to the client and replacing the selected components for which the cryptographic digests do not match; and computing the cryptographic digests on the client to ensure that the downloaded components are authentic and that the components have not been corrupted during transmission.

15. A method as claimed in Claim 1 further comprising:

specifying in the catalog any component of the application program which are no longer needed to execute the program; and

in response to the call to the application program from the client, updating the application program by downloading the catalog from the server to the client and deleting the selected components on the client that are no longer needed prior to executing the updated program.

16. A method as claimed in Claim 1 further comprising specifying a time interval in the catalog, and the application program is updated only on a first time the application is run in the specified time interval.

17. A method as claimed in Claim 16 wherein the time interval is specified by a user from the client.

18. A method as claimed in Claim 1 further comprising specifying a fixed time interval in the catalog at which the application program is updated regardless of whether the call to the server is made from the client.

19. A method as claimed in Claim 18 wherein the application program is updated at the fixed time interval only as necessary to maintain the application program current without executing the application.

20. A method as claimed in Claim 1 wherein the application program on the client is automatically updated on the client each time the client is booted up.

21. A method as claimed in Claim 1 further comprising recording in a file the status of each updating of the application program including names of the components replaced, deleted or added on the client and related procedures for tracking and reporting the program updates.

22. A method as claimed in Claim 1 wherein the components include executable codes, library files, parameter files, and data files of the application program.

23. A method as claimed in Claim 1 wherein the network is the Internet, the server is an Internet server, and the client is a World Wide Web browser, and the call to the server is made through a hypertext link on the browser directed to the catalog on the server.

24. A method as claimed in Claim 23 wherein the Web browser includes a launcher program, the launcher program being executed when the hypertext link to the catalog is selected on the browser, to update the application program and to execute the updated application program thereafter.

25. A method as claimed in Claim 24 wherein the launcher program is a plug-in module to cooperatively run with the browser.

26. A method as claimed in Claim 24 wherein the launcher program is a helper application to cooperatively run with the browser.

27. A method as claimed in Claim 23 further comprising specifying in the catalog a procedure to install an icon on the client which enables a user to subsequently execute the application program without accessing the browser.

28. A method as claimed in claim 23 further comprising specifying in the catalog a procedure to determine whether a launcher program is present on the client, and if the launcher program is not present, retrieving from the server the launcher program and installing the program into the Web browser.

SuB Q3

29. ~~A method as claimed in Claim 1 wherein the network is an intranet and the client is a World Wide Web browser, and the call to the application program is made through a hypertext link on the browser directed to the catalog on the server, the Web browser includes a launcher program, which is executed when the hypertext link to the catalog is selected, to update the application program and to execute the updated application program thereafter.~~

35. ~~30~~. A method as claimed in Claim 1 further comprising specifying in the catalog a procedure to delete the components immediately after executing the updated application to free up disk space on the client.

36. ~~31~~. A system for maintaining an application program on a network comprising:

a server for maintaining the application program, the program including components, each being provided with a version identification;

a catalog on the server for specifying the components with the version identifications;

a client which maintains the application program and, where a user selects the application program to execute the program, first causes the server to download the catalog to the client, compares the version identifications of the components maintained on the server, indicated in the downloaded catalog, and the version identifications of the components maintained on the client, updates the application program on the client by downloading from the server to the client the selected components for which the version identifications do not match and by replacing the selected components on the client, and thereafter executes the updated application program.

37. ~~32~~. A system as claimed in Claim 36 ~~31~~ wherein a portion of the catalog, which includes the components of the updated application program on the client, is stored in a cache on the client.

38. ~~33~~. A system as claimed in Claim 36 ~~31~~ wherein the application program on the client is updated by downloading from the server to the client the selected components not present on the client.

39. ~~34~~. A system as claimed in Claim 36 ~~31~~ wherein the plurality of servers downloads the catalog and components to the client for updating the application program on the client by way of hypertext transfer protocol.

40. ~~35~~. A system as claimed in Claim 36 ~~31~~ wherein the plurality of servers downloads the catalog and components to the client for updating the application program on the client by way of file transfer protocol.

41. ~~36~~. A system as claimed in Claim 36 ~~31~~ wherein the catalog maintained on the server includes network addresses of further servers from which the components can be retrieved.

42. ~~37~~. A system as claimed in Claim 36 ~~31~~ wherein the catalog maintained on the server includes directory locations on the client in which the downloaded components are stored for proper execution of the application program.

43. ~~38~~. A system as claimed in Claim 36 ~~31~~ further comprising a launcher on the client to operate as a proxy to the application program such that when the user selects the application program from the client to execute the program, the launcher is engaged first to communicate with the server to cause the server to download the catalog to the client, update the application program on the client, and execute the updated application program on the client.

44. ~~39~~. A system as claimed in Claim 43 ~~38~~ wherein the launcher is a functional component of an operating system running on the client.

45. ~~40~~. A system as claimed in Claim 43 ~~38~~ wherein the launcher is a stand-alone program on the client for updating the application programs on the client.

46. ~~41~~. A system as claimed in Claim 36 ~~31~~ wherein the catalog includes procedures for executing supplemental programs on the client prior to executing the updated application program.

~~42~~ 47. A system as claimed in Claim ~~41~~ 46 wherein the supplemental programs are executed following the execution of the application program.

~~43~~ 48. A system as claimed in Claim ~~41~~ 46 wherein the supplemental programs include a virus scanning program.

~~44~~ 49. A system as claimed in Claim ~~31~~ 36 wherein the catalog further includes a maximum time interval to limit any delay associated with updating the application program, and a list of further servers on the network, each further server including a copy of the catalog such that when the server fails to download the catalog within the maximum wait-time interval, the client cancels the download and routes the call to one of the further servers to engage a new download and so on until the catalog has been downloaded within the specified maximum wait-time interval.

~~45~~ 50. A system as claimed in Claim ~~31~~ 36 wherein the catalog specifies a cryptographic digest for each component to ensure authenticity and integrity of the component, and the client updates the application program on the client by downloading from the server and replacing the selected components of the application program for which the cryptographic digests do not match, and computing the cryptographic digests on the client to ensure that the downloaded components are authentic and that the components have not been corrupted during transmission.

~~46~~ 51. A system as claimed in Claim ~~31~~ 36 wherein the catalog includes a list of any components of the application program which are no longer needed to execute the program so that, in response to a call to the application program on the client to execute the program, the client updates the application program by downloading the catalog from the server to the client and deleting the selected components on the client which are no longer needed prior to executing the updated program.

52. ~~47.~~ A system as claimed in Claim 36 ~~31~~ wherein the catalog includes a time interval and the application program is updated only on a first time the application is run in the specified time interval.

53. ~~48.~~ A system as claimed in Claim 52 ~~47~~ wherein the predefined time interval is specified by a user from the client.

54. ~~49.~~ A system as claimed in Claim 36 ~~31~~ wherein the catalog includes a fixed time interval at which the application program is updated regardless of whether a call to the application program is made from the client.

55. ~~50.~~ A system as claimed in Claim 54 ~~49~~ wherein the application program is updated at the fixed time interval only if an update is necessary to maintain the application program current without executing the application.

56. ~~51.~~ A system as claimed in Claim 36 ~~31~~ wherein the application program on the client is automatically updated by an operating system on the client each time the client is booted up.

~~52.~~ 57. A system as claimed in Claim ~~31~~ 36 further comprising a file on the client to record status of each updating of the application program, the status including names of the components replaced, deleted or added on the client and related procedures for purpose of tracking and reporting the program updates.

~~53.~~ 58. A system as claimed in Claim ~~31~~ 36 wherein the components include executable codes, library files, parameter files, and data files of the application program.

~~54.~~ 59. A system as claimed in Claim ~~31~~ 36 wherein the network is the Internet, the server is an Internet server, and the client is a World Wide Web browser, and the user selects the application program for execution through a hypertext link on the browser directed to the catalog on the server and procedures in the browser for engaging a download of the components and execution of the program thereafter.

~~55.~~ 60. A system as claimed in Claim ~~54~~ 59 further comprising a launcher program installed on the Web browser, the launcher program being executed when the hypertext link is selected on the browser to update the application program and to execute the updated application program thereafter.

~~56.~~ 61. A system as claimed in Claim ~~54~~ 59 further wherein the catalog is specified with a procedure to install an icon on the client which enables a user to subsequently execute the application program without accessing the browser.

62. ~~57.~~ A system as claimed in Claim ~~54~~ 59 wherein the catalog is specified with a procedure to determine whether a launcher program is present on the client, and if the launcher program is not present, retrieving from the server the launcher program and installing the program into the Web browser.

63. ~~58.~~ A system as claimed in Claim ~~31~~ 36 wherein the network is an intranet and the client is a World Wide Web browser, and the user selects the application program for execution through a hypertext link on the browser directed to the catalog on the server, the Web browser including a launcher program, which is executed when the hypertext link to the catalog is selected, to update the application program and to execute the updated application program thereafter.

64. ~~59.~~ A system as claimed in Claim ~~31~~ 36 wherein the catalog is specified with a procedure to delete the components immediately after executing the updated application to free up disk space on the client.

Sub a4

~~60. A programmed data processing client comprising:~~

~~an application program including components, each being provided with a version identification; and~~

~~a launcher program which, when a user selects the application program to execute the program:~~

~~causes server to download a catalog to the client, the catalog specifying the components with the version identifications;~~

~~compares the version identifications of the components indicated in the downloaded catalog with version identifications of the components maintained on the client; and~~

~~updates the application program on the client by downloading from the server to the client the selected components for which the version identifications do not match and by replacing the selected components on the client.~~

~~61. A program on a storage device providing instructions for execution on a client, which:~~

~~when a user selects to execute an application program, on the client, the application program including components, each component having a version identification, cause a server to download a catalog of components with the version identifications;~~

~~compare the version identifications indicated in the catalog with the version identifications of the components maintained on the client; and~~

~~update the components on the client by downloading from the server to the client the selected components for which the version identifications do not match and replacing the selected components on the client.~~

67. ~~62.~~ A method of installing and automatically updating application programs on a plurality of client computers attached to a common network comprising the steps of:

storing various components of the application programs on one or more server computers attached to the same network with each server operating with standard protocol to automatically transmit a specified file in response to a standard file transfer request;

creating a catalog file which lists the names of all the required components of each of said application programs, and specifying for each component a current version identification and either a content of the component or a network address from which the component can be retrieved by the standard file transfer request;

storing the catalog file on one or more of said server computers; and

installing on each client computer a launcher program which operates as a proxy for each of said application programs and which executes steps for each application program comprising:

retrieving the current version of said catalog file and comparing the components and their version identifications to corresponding information in a second catalog of application components already stored on the client computer;

retrieving from their designated network addresses any components which said launcher program determines as either not present or having incorrect version identifications;

installing the retrieved components in a standard program component directory;

storing the retrieved catalog file to identified the components present on the client in a subsequent update; and

executing the application program.

68. ~~63~~. A method as claimed in Claim ~~62~~ 67 further comprising:

creating on a World Wide Web site a link on a Web page to the catalog file; and

installing on each client computer the launcher program configured as a helper application in a Web browser to automatically execute whenever the link is selected to retrieve the catalog file.

69. ~~64~~. A method as claimed in Claim ~~62~~ 67 wherein the launcher program is configured as a Web plug-in module.

Sub a5

~~65. A method of maintaining an application program in a client-server environment comprising:~~

~~maintaining on a server the application program, the program including the components, each having a version identification, and maintaining a catalog of components with the version identifications;~~

~~in response to a first call to the server from a client, causing the server to download the catalog to the client and the application program;~~

~~maintaining on the client the application program and information in the downloaded catalog including a list of the components with the version idendifications;~~

~~in response to a subsequent call to the server from the client, causing the server to download a second catalog including the latest version identifications of the components and of any new additional components on the server;~~

~~comparing in the client the latest version identifications of the components in the second catalog with the version identifications of the components maintained on the client;~~

~~updating the application program on the client by downloading from the server to the client the selected components for which the version identifications do not match; and~~

~~executing the updated application program on the client.~~

ADD a6

32